



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/066,475	02/01/2002	Edward Colles Nevill	1103179-0009	5943
7470	7590	03/24/2010		
WHITE & CASE LLP PATENT DEPARTMENT 1155 AVENUE OF THE AMERICAS NEW YORK, NY 10036			EXAMINER COULTER, KENNETH R	
			ART UNIT 2454	PAPER NUMBER
			MAIL DATE 03/24/2010	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

RECORD OF ORAL HEARING

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte EDWARD COLLES NEVILL

Appeal 2009-014323
Application 10/066,475
Technology Center 2400

Oral Hearing Held: March 3, 2010

Before ROBERT E. NAPPI, JOHN A. JEFFERY, and
ST. JOHN COURTENAY III, *Administrative Patent Judges*.

APPEARANCES:

ON BEHALF OF THE APPELLANT:

KEVIN X. MCGANN, ESQUIRE
White & Case, LLP
1155 Avenue of the Americas
New York, New York 10036-2787

1 The above-entitled matter came on for hearing on Wednesday, March
2 3, 2010, commencing at 9:00 a.m., at the U.S. Patent and Trademark Office,
3 600 Dulany Street, Alexandria, Virginia, before Paula Lowery, Notary
4 Public.

5 THE CLERK: Good morning. Calendar Number 1, Mr. McGann.

6 JUDGE NAPPI: Good morning.

7 MR. MCGANN: First of all, thank you for taking the time to hear me this
8 morning. The reason we've asked for oral argument on this matter is it
9 appears to me, looking at the record, that there's a fundamental
10 misunderstanding of the Larsen reference and how it works, and how it's
11 being applied to the Applicant's claims.

12 The Applicant's claims are directed to an apparatus and method for
13 microprocessors that would handle multiple instruction sets. Instruction sets
14 are like binary codes that are used and decoded by the processor to tell the
15 rest of the processor what to do.

16 So when you have more than one instruction set, you can imagine that you
17 have essentially two languages or two sets of codes; and the processor needs
18 to know what language or what code it's dealing with -- what instruction set.
19 If it doesn't know the code, it would be like a cryptographer that you give a
20 coded message but you don't tell them the code. They need to know the
21 code to decode it with. So that's why the processor needs to know what
22 instruction set it's dealing with.

23 JUDGE COURTENAY: So you use a high-order bit to designate one of
24 these instruction sets?

25 MR. MCGANN: One example in the Applicant's disclosure is to use a high-
26 order bit that is reserved that does not use an address.

1 The main difference to focus on the Larsen versus the Applicant's invention
2 is that Larsen relies upon, and we can see this -- I might direct you to Figure
3 2 in Larsen, for example, which I've blown up on the second set of materials
4 I handed you.

5 Larsen relies upon this idea of partitioning memory so that the high-order
6 addresses, or the high-order section of I-Store that's there in Figure 2, which
7 is where instructions are stored, has one type of instructions, Type 2
8 instructions.

9 The other portion of memory has a different type of instructions, Type 1
10 instructions, okay?

11 So the gist of what Larsen is talking about is I've partitioned memory. I've
12 separated this out so that I know when I draw instructions from Sector 1
13 memory that they're going to be instruction Type 2. When I draw them from
14 anywhere else in memory, they'll be instruction Type 1.

15 To do that, Larsen provides part of the address -- at least part of the address -
16 - of the instruction. It has a location in memory. It has an address just like
17 our houses have an address. If I want to know where somebody lives or
18 where an instruction resides, I give the address.

19 JUDGE COURTENAY: So the upper three-most bits are used to designate
20 the eight partitions in Larsen?

21 MR. MCGANN: It could be partitioned in any number of ways.

22 JUDGE COURTENAY: Well, they have binary zero to seven here, so it's
23 eight partitions.

24 MR. MCGANN: Correct.

25 JUDGE COURTENAY: We understand that.

26 MR. MCGANN: Okay. If I need to go back to any -- I wasn't familiar with

1 the Court's understanding of technology, so I figured I'd provide a little
2 background. If I jump ahead and you'd like explanation, please stop me.
3 Again, let's look at Figure 2 and look at how this is described in Larsen. The
4 portion that the Examiner cites is in Column 5, line 52 to Column 6, line 16,
5 that I point out in that first slide.

6 The section describes how instructions are decoded, and it makes it clear that
7 the identity of the instruction set is necessary to properly decode the
8 instruction set.

9 In order to identify the type of instruction, I need to use its address.

10 So necessarily in Larsen there are bits from the instruction address that are
11 used to identify the instruction sets to specify how to decode the instructions.

12 We can see that in the section the Examiner has cited against the Applicant's
13 claims where it says the contents of the IDSR-10, which includes part of the
14 address, are taken together with the instruction to provide input to the
15 instruction decoding logic, which is the IDM-5.

16 So the decoding is always done using part of the address and the instruction
17 itself. Without knowing where the instruction came from in Larsen, you
18 can't decode it properly.

19 If you look at Figure 2, for example, and I tried to blow up a particular
20 portion of Larsen Figure 2, and this is -- it has the blue arrows on it.

21 If we look, Larsen explains that the instruction address resides in the IAR
22 register. So that's Item 3 in Figure 2. That instruction address is provided to
23 whatever logic, it's not really described, but logic that would access the
24 instruction storage area on the instruction address box, that's the IAP.

25 So on the left side of Figure 2, we see the instruction address register sends

1 the address to the instruction address plus. That's the whole address -- all of
2 it. That goes up to the I-Store 2 to tell you where the instruction resides.
3 Also, at least a portion of that is reused in the IDSR, that's 10 on Figure 2.
4 So you have this -- and this is what we tried to summarize in the first slide
5 shown in our Brief.

6 JUDGE NAPPI: Stop for a moment. Is that portion of the address still an
7 address?

8 MR. MCGANN: Which portion?

9 JUDGE NAPPI: You said it takes at least a portion of the address to --

10 MR. MCGANN: It is.

11 JUDGE NAPPI: So a portion of an address is still an address?

12 MR. MCGANN: I'm sorry, no, it's not. I misunderstood your question.

13 JUDGE COURTENAY: Why isn't it?

14 MR. MCGANN: Because it wouldn't specify the unique location. Let me
15 give you an example.

16 JUDGE COURTENAY: Okay.

17 MR. MCGANN: My house address is 43. What I've effectively done here
18 is by taking the high-order bits, it would be telling you my address is 4
19 something.

20 JUDGE COURTENAY: Or it could be your address is 43 Robins Lane,
21 Vienna, Virginia, and if I just give you 43 Robins Lane and you happen to
22 already know you're in Vienna, Virginia, that would be a sufficient address
23 to get me to your house, wouldn't it?

24 Your complete address would include your house, the street, and the state.
25 A short address or portion of the address might just be your house number
26 and your street. Why doesn't that same kind of rationale apply to

1 Larsen?

2 MR. MCGANN: Because when I'm dealing with an instruction, I need to
3 know one unique location to pull the exact correct instruction. It may be a
4 portion of the address, as Your Honor mentioned, but it's just that. It can
5 only be a portion.

6 An example in Larsen Figure 2 -- if I have 111, that tells me I'm somewhere
7 in the upper eighth of the instructions.

8 JUDGE COURTENAY: Yes, and then the second part of that address tells
9 you where to look in that I-Store, right?

10 MR. MCGANN: Right, but it would be used together to uniquely identify
11 one spot in the instructions.

12 JUDGE JEFFERY: I think the Examiner is taking the position that the last
13 five bits, if you will, of this eight-bit address in the I-Store corresponds to
14 the subset -- I'm particularly referring to Claim 65, by the way.

15 MR. MCGANN: Okay.

16 JUDGE JEFFERY: It's probably best to jump to the broadest claim first.
17 The subset that's being recited here, as I understand the Examiner's position,
18 the Examiner is taking the position that the last five bits correspond to that.
19 As I understand your position in the Reply Brief on page 8, you indicate
20 those bits only indicate part of an address, not the entire address, and there's
21 eight possible addresses, in effect, that could be identified here by the top
22 three bits.

23 So you have the same last five bits and three different top three bits, you're
24 going to get eight different addresses identified is what I understand. So
25 your position is that because there are eight possible combinations of

1 addresses in taking the Examiner's position, you couldn't possibly decode the
2 correct address, is my understanding.

3 MR. MCGANN: I think Your Honor has said and as I said, you would not
4 uniquely identify an address in this example where you have it divided in
5 eighths. You'd have one of eight possibilities.

6 JUDGE JEFFERY: One of eight possibilities and when you go to the claim,
7 the claim simply says "an ordered set of bits" -- Claim 65 -- "wherein a
8 subset of the ordered set of bits identifies . . .," and I'm going to emphasize
9 this word, "*an* address of *an* instruction." Does that preclude an *incorrect*
10 address of an *incorrect* instruction?

11 MR. MCGANN: I wouldn't think that somebody would read it that way to
12 include an incorrect address because then it wouldn't function.

13 If I continually access the wrong instructions, as we pointed out in the Reply
14 Brief, just picking one out of eight, you'd be wrong 90 percent of the time.
15 You'd be wrong 90 percent of the time by design, not to mention if there
16 were any errors in actual operation.

17 I would assume that somebody skilled in the art would not read a claim in a
18 way that would fail 90 percent of the time.

19 JUDGE JEFFERY: So you have a one in eight chance of getting it right,
20 basically, taking this Examiner's position with the last five bits being the
21 subset.

22 I would also add that Claim 65 doesn't have any functionality, does it? I
23 mean, we're not doing anything with these bits, unlike some of the other
24 claims where we're using the information to decode or the like. Here, we're
25 just identifying things, are we not?

1 So my point being: the fact that the Examiner's position as to the last five
2 bits -- certainly in the Examiner's mind, these things are identifying -- these
3 last five bits are identifying some type of instruction, albeit partial.

4 MR. MCGANN: I guess my response would be that they don't identify an
5 instruction.

6 JUDGE JEFFERY: An address --

7 MR. MCGANN: They would identify a range of possible instructions.

8 JUDGE JEFFERY: One of eight.

9 MR. MCGANN: Which makes them not be an address. That subset isn't an
10 address because it doesn't tell you where to go to get that instruction.

11 JUDGE COURTENAY: Actually, in the reference they're referred to
12 specifically as high-order addresses at Column 5, Line 47. It says these are
13 identified as having the high-order address beginning with "111" in binary.
14 So they're specifically referred to in the reference as high-order addresses --
15 the upper most three-fifths -- that identify the eight partitions.

16 MR. MCGANN: That, in my view, exactly supports the Applicant's
17 argument. Those bits are required as part of the address.

18 JUDGE COURTENAY: So it's by virtue of identifying the different
19 partitions -- the eight partitions in Figure 2 -- it illustrates that the upper-
20 most partition, Partition 7 in binary "111," is reserved for Type 2
21 instructions. The rest of the partitions are Type 1 instructions.
22 It also identifies the type of instruction set.

23 MR. MCGANN: That's exactly the problem I have with applying Larsen to
24 the Applicant's claims. I think in one or more ways each of these claims
25 requires that you can't use the address, or part of the address, to specify the
26 instruction set or instruction type; and Larsen can't do that.

1 JUDGE COURTENAY: Can you point to your language in Claim 65 that
2 precludes that?

3 MR. MCGANN: Sure. In Claim 65 we have -- it refers to what would be in
4 the program counter. You have the ordered set of bits, okay? There's a
5 subset of that ordered set that identifies the address of the instruction, okay?
6 There's at least one bit in that set that identifies the instruction set. The last
7 part of that is the important part where at least one bit is not a member of the
8 subset.

9 If we look at Larsen, you have these eight bits that are all the address. You
10 have three of those eight are repeated and used again to identify the
11 instruction set. So there is

12 no -- there can't be at least one bit that identifies the instruction set, meaning
13 those three, that are both part of the address and identify the instructions.

14 JUDGE JEFFERY: The Examiner is simply taking the position that the top
15 three bits plus the five -- all eight of the bits, the so-called "ordered set of
16 bits," and that the subset in the claim is the last five bits.

17 So the top three clearly indicate the type of instruction that you have in
18 terms of the partition that you're in in the I-Store, whether it's a "111" or
19 what have you, but the last five bits in isolation, in the Examiner's mind,
20 seem to indicate an address, albeit partial.

21 MR. MCGANN: I guess my response would be that is, as you said, a partial
22 address. It's not an address. I think the claim says that subset has to identify
23 an address.

24 JUDGE NAPPI: Counsel, it seems to me we're arguing over what the
25 definition of an address is. You seem to be saying an address is complete,

1 all the information you need; and we're saying any information that gets you
2 to a location.

3 What I'd like to know, since we're looking at the claims now, where is your
4 definition of address in your specification to find that it's got to be in such a
5 manner that precludes what the Examiner has said?

6 MR. MCGANN: Well, one example --

7 JUDGE NAPPI: A definition.

8 MR. MCGANN: A definition?

9 JUDGE NAPPI: Yeah, I think we're arguing back and forth over examples.
10 I'd like to see a definition that nails it.

11 MR. MCGANN: One example might be in Column 4, beginning around
12 Line 5 -- actually, it's line 6. It refers to the program counter register.

13 JUDGE NAPPI: Um-hum.

14 MR. MCGANN: The program counter register addresses memory.

15 JUDGE NAPPI: Okay.

16 MR. MCGANN: So the address that it would use is the information that
17 would be residing in the program counter.

18 If you look at Column 4 along side of Figure 1 of the Applicant's, you see
19 it's Item 130, the PC? That's the portion of that that other than the part that's
20 labeled T, the portion of that is what would contain the address.

21 JUDGE NAPPI: Okay. You have just given us this nice figure of Larsen,
22 which has your portion of T-1 and PC together on the right-hand side, and
23 then Larsen that has the high-order bits and lower bits.

24 Again, we keep coming back to the same issues. We have examples here,
25 and the example in Larsen -- it seems to me the Examiner is saying that the
26 low-order address is an address for a certain portion of memory.

1 You're saying, well, that's not the complete address, and I'm trying to
2 understand why we have to interpret your claim as being to the complete
3 address.

4 MR. MCGANN: That section in Column 4 refers to the program -- I'm
5 looking at approximately Line 8, where it talks about program counter
6 controller increments the value in the program counter. As each instruction
7 is executed, a new instruction must be fetched.

8 JUDGE NAPPI: Okay.

9 MR. MCGANN: So the address is to fetch an instruction.

10 JUDGE NAPPI: Okay.

11 MR. MCGANN: It's not to point to a general 1/8th, for example, of the
12 addressable memory.

13 JUDGE NAPPI: But is it fetching the instruction?

14 MR. MCGANN: I'm sorry?

15 JUDGE NAPPI: In Larsen, is it fetching the instruction in that 1/8th of the
16 memory?

17 MR. MCGANN: Not without using those three high-order bits. It wouldn't
18 fetch the instruction.

19 JUDGE NAPPI: Okay.

20 MR. MCGANN: It couldn't fetch an instruction without using those three
21 high-order bits, which is why it never -- the whole point of Larsen is it relies
22 upon partition in the memory so that address tells you -- and this is even in
23 the section that the Examiner cites to us.

24 I'm reading from Column 5, and this is -- the section the Examiner cites is
25 Line 52 to Column 6, Line 16 of Larsen. I'm reading particularly at Column
26 6, Lines 11-14.

1 It says: "The decoding of the instruction depends not only the contents of
2 the IDR-1, which is the instruction itself, but on the region of the I-Store
3 from which the instruction was fetched."

4 So that region, which is specified by part of the address, is used twice. It's
5 used to give you the address -- 111 followed by five or six digits that
6 specifies the address.

7 JUDGE NAPPI: But doesn't that section also say that the low-order bit is
8 used to fetch an instruction from IDR-1?

9 MR. MCGANN: I'm sorry, where are you referring to?

10 JUDGE NAPPI: Once you're in IDR-1, doesn't the low-order bit identify an
11 instruction to be fetched from IDR-1?

12 MR. MCGANN: IDR-1 in Larsen actually contains the instruction itself. I-
13 Store contains the --

14 JUDGE JEFFERY: Yeah, there's two things. The address is stored in the I-
15 Store, and the instruction is stored in the IDR. The actual instruction itself is
16 stored in the IDR.

17 MR. MCGANN: Correct.

18 JUDGE JEFFERY: Right. So the question is, fetching an instruction: Is
19 that dependent on the address of the instruction?

20 MR. MCGANN: It would be because you would have -- think of the
21 memory that you use. This could be many megabits or more of memory.
22 You need to tell the controller where to go in that memory to get the
23 instruction you want to execute.

24 There might be -- the example of pipelining, where you have multiple
25 instructions being fetched at one time, but it would still be a specific set or
26 address -- a specific range in the I-Store -- that you want to address.

1 JUDGE JEFFERY: Counsel, I wanted to ask you -- our time is limited here.
2 I do want to ask you about one thing that caught my eye in Column 6 of
3 Larsen, you know, beginning around Line 3, where it talks about the content
4 of the IDSR-10 and the IDR-1 are taken together to provide a look-up
5 address within the IDM-5. Fine.

6 This next couple of sentences though caught my eye. "Only specific portions
7 of the undecoded instruction in IDR may be needed according to the format
8 of the instruction being fetched. The other portion of the input address to the
9 IDM is the output from the IDSR-10, which identifies the fetch-from address
10 in the I-Store."

11 What does that mean? Only specific portions of the undecoded instruction
12 in the IDR may be needed? Are we talking -- does that bolster the
13 Examiner's position some how some way in terms of just needing the last
14 five bits, if you will? It's not the full number of bits?

15 MR. MCGANN: With all due respect, that's the instruction, not its address.

16 JUDGE JEFFERY: Okay.

17 MR. MCGANN: The Examiner's argument is that I can use some part of an
18 address, and it's an address. I think the response is it's not an address in the
19 instruction.

20 JUDGE JEFFERY: Okay.

21 MR. MCGANN: When you have an instruction, you have -- separately,
22 they're all ones and zeros, but you have parts of the instruction that tell the
23 processor what to do. Other parts may tell is what the operands are. So the
24 instruction is: and the operands are 3 and 4.

25 JUDGE JEFFERY: Why do you think Larsen said the other portion of the
26 input address right after that sentence? I know the preceding sentence says

1 "... and portions of the instruction," so that would be the instruction itself,
2 but then he goes on to say "... the other portions of the input address." Is
3 that just an anomaly in the wording?

4 MR. MCGANN: It's a specific way the implementation of Larsen -- instead
5 of having --

6 JUDGE JEFFERY: Why did he call it "the other portions"? That's my
7 question.

8 MR. MCGANN: The way Larsen is set up with the IDM-5
9 is -- you have two ways to decode an instruction, a least two ways.

10 One is hard wire the logic, ands and operands where I apply electrical
11 signals to it and I get a definite output.

12 Another is I have a look-up table where the bits of the instruction -- not the
13 address, the bits of the instruction -- that tell you it's an add instruction. Tell
14 me go to this place in the instruction decoder to determine what to do.

15 That's completely separate from where the instruction is stored in I-Store
16 and memory. It's a function of the specific implementation.

17 JUDGE JEFFERY: So the terminology "the other portion of the input
18 address" to the IDM-5, "the other portion of the input address" would be the
19 top three bits? Is that what we're talking about?

20 MR. MCGANN: No.

21 JUDGE JEFFERY: The "other portion" --

22 MR. MCGANN: It's not an address that's used to address the instruction at
23 all. It's a look-up table --

24 JUDGE JEFFERY: Well, it says the output of the IDSR-10. Isn't that the
25 top three bits?

26 MR. MCGANN: Oh, I'm sorry. The top three bits of the IDSR --

1 JUDGE JEFFERY: Right, let's make sure we're on the same sheet of music
2 here. I'm starting on Line 8 of Column 6, and I'll quote:

3 "The other portion of the input address to the IDM-5 is the output from the
4 IDSR-10 which identifies the fetched-from address."

5 That would be the top three bits, would it not?

6 MR. MCGANN: Yes.

7 JUDGE JEFFERY: Okay. The only reason I bring this up is that
8 terminology "other portion of the input address" comes right after where he
9 talks about specific portions of the undecoded instruction. Putting those two
10 things together, it sounds like one portion could be the last five bits, and the
11 other portion, the top three bits.

12 What I think I hear you saying is, they're apples and oranges, in effect.

13 You've got addresses, you have instructions, portions of each aren't the
14 same.

15 MR. MCGANN: You do, and if I could beg your indulgence for two
16 minutes to try to explain what Larsen Figure 2 shows, maybe I could clear
17 that up.

18 JUDGE JEFFERY: Well, I understand how this works. I don't know if that
19 particular passage is reflected in the functionality here specifically.

20 You know, we're taking address information along with the instruction itself
21 and using that as a look up in the instruction decode memory to decode the
22 instructions. So I think we all understand how this works.

23 MR. MCGANN: Right.

24 JUDGE JEFFERY: It was sort of a peculiar terminology, I thought, in the
25 reference that it was talking about "the other portions" kind of thing.

1 MR. MCGANN: Well, I think that may also suggest -- although it's in a
2 different context because it's in decoding, the address really refers to what
3 would be a unique location, rather than a region.

4 JUDGE JEFFERY: I have one other question for you, if I may. There
5 seems to be, from the record reading this over, a bit of inconsistency in how
6 the Examiner is interpreting -- I'm going to Claim 1 now, the last clause of
7 Claim 1 -- the memory access controller.

8 MR. MCGANN: Yes.

9 JUDGE JEFFERY: Frankly, the Examiner's rejection on page 9 of the
10 answer is a little tough to sort out exactly how the Examiner is mapping the
11 elements to the memory address controller. Specifically in Item (e) on page
12 9 of the Answer, he refers to Figure 2 and then cites three different elements:
13 3, 10 and 2, which are the IAR, IDSR, and the I-Store, respectively, there.
14 In your Brief on page 18 of your Brief, you tend to refer to the memory
15 access controller, line 9 on page 18 of the Brief, you say: "Memory Access
16 Controller, the address-resolving logic 6, in Larsen."
17 So do we have an inconsistency there in terms of how we're interpreting that
18 element?

19 MR. MCGANN: I believe my reference to the address was on the logic 6,
20 and that on page 18 you referred to was incorrect.
21 It is, in fact, not really shown in the Larsen figure. It's whatever logic would
22 use the address in the I-Store, not decoding logic.

23 JUDGE JEFFERY: All right.

24 MR. MCGANN: I apologize, that was my mistake. I apologize for the
25 confusion.

26 JUDGE JEFFERY: Okay.

1 MR. MCGANN: I appreciate you pointing out “(e)” because I think it's not
2 just an inconsistency. I think the Examiner has misunderstood that whatever
3 memory access controller there is, the access is the I-Store.

4 The Examiner in responding to (e), Claim 1, says those bits in the top three
5 bits, the access controller would not be responsive to that; and that's clearly
6 not the case in Larsen. As we see in Figure 2 and the description, all the bits
7 of the address are put on the address box.

8 For example, if we look at Figure 2, we have the instruction addresses in the
9 IAR-3. Those bits -- all of them -- go up to the IAD to the I-Store.

10 So those bits are not the controller that controls how to decode the
11 instruction -- I'm sorry, how to retrieve the instruction from memory. It can't
12 be not responsive to those bits. It's definitely responsive. Those three bits
13 are included in the address. So Larsen couldn't disclose that.

14 JUDGE COURTENAY: But isn't it true once you have the partition
15 selection by the high-order three bits, you have eight partitions of
16 instructions? The upper-most partition is the Type-2 instructions, and the
17 rest are Type 1.

18 Once you have a given partition selected, the low-order address bits
19 determine an instruction in a given partition, isn't that true? Would you
20 agree to that?

21 The low-order address bits uniquely address an instruction within a given
22 partition that has been selected by the high-order bits.

23 MR. MCGANN: But it can only be done in combination with those high-
24 order bits. I think even as Your Honor just phrased it, once those are
25 identified by the high-order bits, the high-order and low-order bits are used
26 together to determine the address.

1 JUDGE COURTENAY: Well, it seems your position is the subset of the
2 ordered set of bits, being the lower order address bits, doesn't uniquely
3 identify an address.

4 MR. MCGANN: Okay.

5 JUDGE COURTENAY: But it does within the context of a given partition.
6 The low-order address bits uniquely identify an address, an instruction, once
7 the partition has been selected by the high-order address bits.
8 The high-order address bits have a dual function. They select the partition,
9 but they also, by selecting the different partitions, they identify the type of
10 instruction.

11 MR. MCGANN: I agree with the latter part of what Your Honor said. The
12 high-order address bits -- because you call them address bits -- they are the
13 high-order address bits, do both functions.

14 JUDGE COURTENAY: They have dual functions as I see it.

15 MR. MCGANN: I'm sorry?

16 JUDGE COURTENAY: They have a dual function.

17 MR. MCGANN: Correct, and that's exactly what we're saying distinguishes
18 the Applicant's invention. Because if you look, for example, at Figure 2, I
19 believe it is of the Applicant's invention, that may be the best place it's
20 illustrated -- Figures 2 and 3.

21 You see it refers to the memory address and then separately this bit that tells
22 you what the instructions are. So you have an address, and you have
23 something else that tells you the instruction type.

24 You don't need to reuse -- this provides a non-trivial advantage in
25 application. You don't have to partition memory because I don't care where
26 the instructions of different types are stored.

1 It would be important to a software designer. Imagine if I had partitioned
2 1/8th of the memory, and the instructions that I wanted to use of that
3 particular type spilled over?

4 JUDGE COURTENAY: I understand the advantage you're discussing, but
5 we're interested in what the broadest reasonable construction of your claim
6 reads on the reference. That's our concern.

7 MR. MCGANN: I think the broadest reasonable construction couldn't read
8 on it because an address of an instruction is its location in memory. It's a
9 place, not a region.

10 JUDGE COURTENAY: Don't you agree we have a partial address in these
11 low-order address bits that uniquely identify an instruction in a given
12 partition?

13 MR. MCGANN: They provide a partial address.

14 JUDGE COURTENAY: A partial address is an address. If you consider an
15 address the genus and you consider a partial address the species, a species
16 anticipates a genus.

17 MR. MCGANN: But they wouldn't identify the address of an instruction.

18 JUDGE COURTENAY: Within a given partition it would. Once that
19 partition is selected by the higher order address bits.

20 MR. MCGANN: But that very point is you have to use those high-order
21 address bits to know the region and combine -- so the address is the
22 combination of those bits.

23 It has to include the high-order bits, otherwise I could be in any position in
24 the memory.

25 JUDGE COURTENAY: That depends on how we construe an address.

26 Whether an address is just an address within a given partition is sufficient to

1 meet your claim, or whether we have to look at all the address bits as
2 uniquely identifying a given location of an instruction within all eight
3 partitions.

4 MR. MCGANN: Well, I think the fair reading of the claims is it identifies
5 an address and instruction. Taking Your Honor's point as to 65, which is the
6 first one the Examiner dealt with, the language of the claim says the ordered
7 sets of bits identifies an address and instruction, not a regional I-Store, not
8 one of many possible instructions. It's an address.

9 JUDGE COURTENAY: But again, the low-order bits do identify an
10 address of an instruction within the context of a single function.

11 MR. MCGANN: Not alone. They can't do it alone because built into your
12 very question, Your Honor –

13
14 JUDGE COURTENAY: I agree that the partition has to be selected by the
15 high-order three bits.

16 MR. MCGANN: So to go back to Judge Nappi's example, if I had a house
17 number and street name in the city, the address of my house is all of those
18 things.

19 The address of an instruction -- what I understand Your Honor is saying, if I
20 knew, for example, that I was in Alexandria, Virginia, and knew further
21 what neighborhood in Alexandria I was in, I might be able to give a street
22 number and name and somebody would find the location.

23 But that's not how instructions work. I've already built into that assumption
24 that I know I'm in Alexandria, I know the region I'm in. That's the address.
25 It's what uniquely identifies that building. This building is 600 Dulany
26 Street.

1 JUDGE COURTENAY: Well, if Dulany Street corresponds to the high-
2 order bits, say we have three bits that uniquely identify Dulany Street to use
3 your analogy, 600 would be the low-order bit that uniquely identifies the
4 location within Dulany Street.

5 The 600 Dulany Street is an address and is Dulany Street. It's just a broader
6 address when you're looking at the street, or in the case of the reference the
7 three high-order bits identify a unique partition.

8 MR. MCGANN: Right, I would think of the address in this case and as
9 reflected in the claim language, it's an address of an instruction. The
10 analogy would be it's an address -- it depends on what the address is
11 supposed to be of.

12 If it's a city, Alexandria might be enough. If it's an instruction, the
13 instruction has a specific location in memory that's defined by the high-order
14 bits and the low-order bits in Larsen.

15 JUDGE COURTENAY: So I understand your position to be that an
16 address, as recited in Claim 65, can only be a unique address within the
17 context of all eight partitions in the Larsen reference shown on Figure 2.
18 That's your position, where you need the whole address.

19 MR. MCGANN: I believe that's correct.

20 JUDGE COURTENAY: I understand your position.

21 MR. MCGANN: It has the address of an instruction.

22 JUDGE NAPPI: Any more questions?

23 JUDGE COURTENAY: No.

24 JUDGE JEFFERY: No.

25 JUDGE NAPPI: Thank you very much.

26 Whereupon, the proceedings at 9:40 a.m. were concluded.